INTERPRETABLE MODELS FOR AUTOMATIC SLEEP STAGE SCORING

A Thesis
Presented to
The Academic Faculty

By

Irfan Al-Hussaini

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

May 2020

INTERPRETABLE MODELS FOR AUTOMATIC SLEEP STAGE SCORING

Approved by:

Dr. Jimeng Sun, Advisor
School of Computational Science and Engineering
Georgia Institute of Technology

Dr. Omer T. Inan, Co-advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Justin Romberg
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Christopher J. Rozell
School of Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: April 24, 2020

To Dada, Fupi, and Nanu

ACKNOWLEDGEMENTS

I am thankful to my advisor, Professor Jimeng Sun, for welcoming me to be a part of his dynamic research team, giving me the freedom to pursue independent research, and being a great mentor.

I am very grateful to my collaborators Dr. Brandon Westover and Dr. Cao Xiao, whose invaluable advice helped drive the project along.

I would also like to thank my co-advisor Professor Omer T. Inan, and my committee members, Professor Justin K. Romberg and Professor Christopher J. Rozell for their valuable time and suggestions towards improving the quality of this thesis.

Thanks to my incredible friends who were always by my side, and my colleagues at Sunlab for their insights, talks, and support.

Finally, I am eternally grateful to my parents, my grandfather, my sister, and my brother for giving me everything anyone could ever need.

TABLE OF CONTENTS

Acknowledgments iv

List of Tables vii

List of Figures viii

Chapter 1: Introduction 1

1.1 Sleep Staging 1

1.2 Interpretation Incorporating Domain Knowledge 4

1.3 Technical Significance 5

1.4 Clinical Relevance 6

Chapter 2: Method 7

2.1 Method Overview 7

2.2 Dataset 8

2.3 Expert Rule Embedding 9

2.4 Signal Embedding Generation 13

2.5 Prototype Learning and Relevance Matching 14

Chapter 3: Experiments 16

3.1 Experiment Setting 16

3.2 Results: Staging Performance 18

3.3 Selection of Expert Rules 21

3.4 Interpretation 23

Chapter 4: Conclusion 25

References 29

LIST OF TABLES

2.1 Dataset Summary 9

3.1 Model Evaluation 18

3.2 Sensitivity across Sleep Stages 19

3.3 96 selected rules out of 240 expert rules using ISRUC dataset 22

3.4 96 selected rules out of 240 expert rules using MGH Dataset 22

LIST OF FIGURES

2.1 The SLEEPER framework. 8

2.2 Location of electrodes for EEG in the 10–20 system. 10

2.3 Convolution Layers of CNN 14

2.4 A 30 second epoch having 9 channels 14

3.1 Confusion Matrices on ISRUC dataset 20

3.2 SLEEPER ROC-AUC vs Tree Depth 21

3.3 SLEEPER ROC-AUC vs Number of Rules 21

3.4 SLEEPER-Decision Tree trained from the ISRUC dataset 23

SUMMARY

This thesis aims to combine domain knowledge with deep learning to develop interpretable yet robust models for a particular clinical decision support system, sleep staging. The method is transferable to other areas where domain knowledge can be represented by a set of computational rules. Currently, sleep staging, a cardinal step for evaluating the quality of sleep, is a manual process, done by sleep staging experts who are trained over months. Moreover, it is tedious and complex as it can take the trained expert several hours to annotate just one patient's polysomnogram (PSG) from a single night. As a result, data-driven methods for automating this process have been explored extensively by the research community and deep learning models have demonstrated state-of-the-art performance in automating sleep staging. However, interpretability which defines other desiderata has largely remained unexplored. In this thesis, we propose `SLEEPER`: interpretable Sleep staging via Prototypes from Expert Rules, a method for automating sleep staging which combines deep learning models with expert-defined rules using a prototype learning framework to generate simple interpretable models. It derives a prototype, which is a representative latent embedding of PSG data fragments, for each sleep scoring rule and expert-defined feature. The inference models are simple and interpretable like a shallow decision tree whose nodes are based on a similarity index with those meaningful rules and features. We evaluate the method using two PSG datasets collected from sleep studies and demonstrate that it can provide accurate sleep stage classification comparable to human experts and deep neural networks with about 85% ROC-AUC and .7 κ.

CHAPTER 1
INTRODUCTION

Deep Learning has infiltrated a lot of fields with unprecedented success. However, it often creates "black-box" models that are difficult for a human to clearly understand and create new meaningful knowledge. In important fields like healthcare and medicine interpretability of models can be the difference between adoption and rejection.

So, what defines interpretability? Are linear models, decision trees, and rule lists always more interpretable than neural networks? The answer does not appear to be binary [1]. Propagation to the leaf of a very tall decision tree can be more difficult to follow than a neural network with a couple of layers. Even the hidden layer can elucidate meaningful semantic information to a human such as the meaningful word association identified by word2vec algorithm [2].

The objective of this thesis is to elucidate an interpretable model for automatic sleep staging which is deployable at sleep centers due to a balance of accuracy and human interpretability of the classification algorithm.

1.1 Sleep Staging

Sleep plays a vital role in physical, mental, and social welfare, impacting over 100 million Americans each year [3]. Inadequate or irregular sleep can adversely affect an individual by disrupting memory retention and learning capability [4]. A study of the relationship between moral reasoning and sleep deprivation revealed moral reasoning was substantially impaired during partial sleep deprivation [5]. Sleep disorders such as sleep apnea and insomnia affect over 50 to 70 million US adults, many of whom are undiagnosed [6].

The central diagnostic test is through sleep studies which involve collecting and analyzing polysomnograms (PSG) data of patients during sleep which includes the following

electrophysiological signals: electroencephalogram (EEG), electrooculogram (EOG), electromyogram (EMG), and electrocardiogram (ECG). For sleep analysis, these PSGs are used to characterize each 30 second segment of recording by one of 5 sleep stages: Wake (W), Rapid Eye Movement (REM), Non-REM Stage 1 (N1), Non-REM Stage 2 (N2), and Non-REM Stage 3 (N3). This process is called sleep staging and is the most important step for diagnosing sleep disorders such as insomnia, narcolepsy, or sleep apnea [3]. These annotations are performed according to characteristics defined in *The AASM Manual for the Scoring of Sleep and Associated Events* [7]. Typically neurologists will visually inspect multivariate PSG time series and provide manual scores of sleep stages. Such a visual task is cumbersome and requires sleep experts to manually inspect PSG data recorded during the whole sleep study. So, this annotation phase is an expensive and time-consuming procedure requiring the observation and labeling of each 30 second segment of a recording that can be more than 8 hours long for a single session and it can take several hours to annotate one patient's record during a single night.

To overcome this limitation, there has been a considerable effort over the years to develop deep learning methods to automate the sleep scoring task due to their promising performances. Recent research include developing artificial visual perception using convolutional neural networks (CNN) [8], recurrent neural networks (RNN) [9], recurrent convolutional neural networks (RCNN) [10] and deep belief nets [11]. Although deep learning models can produce accurate sleep staging classification, they are often treated as black-box models that lack interpretability and transparency of their inner working and method of inference [12]. This can limit the adoption of the deep learning models in practice because clinicians often need to understand the reason behind each classification to avoid data noises and unexpected biases.

On the other hand, current clinical practice at sleep labs rely on the American Academy of Sleep Medicine (AASM) sleep scoring manual [7], which are interpretable for clinical experts but often vague and not computationally precise. Furthermore, the real data are

much more heterogeneous and noisy, which leads to more difficult cases to score. As a result, even after certification, technicians often need to acquire multiple years of working experience in scoring real-patient data at sleep labs before their scores can be trusted.

Can we develop models that are as **interpretable** as the sleep scoring manual but as **accurate** as the black-box neural network models? To acquire such a sleep staging model that can produce both accurate and interpretable results, we propose a method based on **prototype learning**, which is an interpretable model inspired by case-based reasoning [13], where observations are classified based on their proximity to a prototype point in the dataset. Many machine learning models have incorporated prototype concepts [14, 15, 16], and learn to compute prototypes (as actual data points or synthetic points) that can represent a set of similar points. These prototypes provide an intuitive understanding of the classifications. Prototype learning also had successes in deep learning models [17, 18]. The challenges of developing prototype learning methods with deep learning include

1. the resulting models are not necessarily interpretable as the final models are often still complex neural networks;

2. those models do not capture existing domain knowledge such as scoring rules from the training manual.

The method we propose, *Sleep staging via Prototypes from Expert Rules* (`SLEEPER`) [19], combines deep learning models with expert-defined rules via a prototype learning framework to generate simple interpretable models such as shallow decision trees and logistic regression models. In particular, `SLEEPER` utilizes sleep scoring rules and expert-defined features to derive prototypes which are embeddings of polysomnogram (PSG) data fragments via convolutional neural networks. The final models are still simple interpretable models like a shallow decision tree or logistic regression defined over those phenotypes.

1.2 Interpretation Incorporating Domain Knowledge

The nuance of interpretability lies two-fold, in the feature and the classifier. To tackle the problem, we extract binary features defined by meaningful rules and use a decision tree as the classifier so predictions can be elucidated by those rules. Due to the ambiguous nature of interpretability [1], we build on rules defined by experts in the field. The official document detailing the consensus rules of sleep staging is *The AASM Manual for the Scoring of Sleep and Associated Events* [7].

However, these rules do not have clearly defined boundaries that would enable computational implementation. Instead, the sleep technicians go into training where the actual rules are learned over time. The ambiguity further explains the disagreement of 18% among the two annotators in the dataset. Due to the difficulty of a complete rule-based system, we augment the set of implementable rules from the official document with an additional set of meaningful rules as suggested by experts in the field. Each prediction from our model is traceable to a subset of these rules and provides a meaningful explanation for each prediction.

The interpretation is derived from a novel combination of CNN embeddings, expert-defined rules, and the resulting prototypes. Each sample is modeled as a multi-hot vector obtained from the expert-defined rules. A prototype is the average of each rule in the high-dimensional space of CNN embeddings. Together, these prototypes represent a mapping from CNN embeddings to rules. This enables the utilization of high-dimensional embeddings to represent each set of electrophysiological signals and map them to a similarity index with each rule. This resulting similarity index is used to make predictions using a decision tree and trace the reasoning with the set of satisfied rules. This model results in an accuracy of 78% which is within 4% of the agreement of two human experts on the same dataset and compares favorably with other models. According to our experiments, having epochs from a PSG in training and testing is favorable for evaluation. This can

be explained by the physiological heterogeneity between patients. Thus, to emulate a real clinical setting, our reported evaluation metrics are based on new PSG recordings absent from the training.

We create a combination of the official attributes used by sleep clinicians to annotate sleep stages. Each rule has a separate methodology requiring the extraction of different discriminatory features such as slow-wave activity, rapid eye movements, and sleep spindles. This rule list is followed by mirroring the AASM Manual for Sleep Staging [7]. We are unable to use all attributes computationally due to the nature of their definition. So, we generate groups of epochs classified by each of these attributes. Samples discriminated by the same rules should have a minimum cosine similarity index with the same prototype. This validates or invalidates the development of clinically relevant rules in the latent space of our system. Although we focus on the sleep staging application, this proposed technique can be extended to a multitude of other use cases to develop interpretable computational methods for classification.

1.3 Technical Significance

Although deep learning models have demonstrated state-of-the-art performance in sleep staging, their interpretability has largely remained unexplored. Interpretability helps determine the extent of desiderata beyond performance metrics such as fairness, privacy, reliability, robustness, causality, usability, and trust [20]. In the current study, to achieve accurate but much more interpretable sleep stage classification, we develop a framework that first jointly embeds both multivariate PSG data and the staging rules followed by experts into the same latent space using CNN, so that relevance scores between each rule and data prototype can be computed using normalized cosine similarity. It then performs staging classification using a decision tree and learns staging rules along with relevant prototypes. The results include both expert rules and PSG prototypes, which mimics the visual inspection mechanism of clinical experts. Moreover, this method can be applied to other

domains to develop interpretable models if the domain knowledge can be represented by a set of rules.

1.4 Clinical Relevance

Dysfunctional sleep can lead to multiple medical conditions including cardiovascular, metabolic, and psychiatric disorders [3]. Sleep deprivation in the form of insomnia affects 10-15% of the adult population causing distress and impairment [21] with effects ranging from poor memory to increased susceptibility to motor vehicle accidents [4]. Sleep staging is the most important precursor to sleep disorder diagnosis. However, manual sleep staging is labor-intensive and expensive. Computational Sleep Stage Scoring can amortize the cost of diagnosing sleep disorders. Although automatic sleep staging has been explored in-depth, interpretation of resulting models remains unexplored. `SLEEPER` provides a set of clinically meaningful phenotypes, for each prediction. The phenotypes, referred to as prototypes, are derived through rules outlined in *The AASM Manual for the Scoring of Sleep and Associated Events* [7] and augmented by suggestions from sleep experts. Our resulting shallow decision trees can potentially enhance the training of sleep technicians to learn complex phenotypes related to sleep stages via intuitive explanation. Moreover, it can reveal the significance of new phenotypes in the classification of particular sleep stages.

CHAPTER 2

METHOD

2.1 Method Overview

SLEEPER identifies sleep stages on PSG data via interpretable classification models over explainable patterns extracted by expert defined rules. The input data are multi-channel PSG signals segmented into 30-second epochs in the form of multivariate continuous time series data, denoted as $\mathcal{X} = \{\boldsymbol{X}_1, \cdots, \boldsymbol{X}_N\}$, where each epoch $\boldsymbol{X}_n \in \mathbb{R}^{9\times 6{,}000}$ is 30 seconds long and contains 9 physiological signals recorded at a frequency of 200Hz. Each epoch $\boldsymbol{X}_n$ has a sleep stage label $y_n \in \{\text{Wake, REM, N1, N2, N3}\}$. Our task is to predict the sequence of sleep stages $\boldsymbol{S} = \{s_1, \ \ldots \ s_N\}$ based on $\mathcal{X}$ so that they are close to the human labels $\boldsymbol{Y} = \{y_1, \ \ldots \ y_N\}$. We also aim at providing explainable predictions using interpretable classifiers, which are enhanced with neural networks and expert defined rules.

As shown in Figure 2.1, SLEEPER comprises of several modules:

- **Signal embedding module:** We begin with training the CNN on the end-to-end task of predicting sleep stages using raw PSG data. Afterwards, we remove the last fully-connected layer of the trained CNN and obtain a latent representation, $\boldsymbol{h}(\boldsymbol{X}_n)$ for epoch n.

- **Expert rule module:** Concurrently, we use a set of expert rules to encode each epoch into a multi-hot vector, $\boldsymbol{R}(\boldsymbol{X}_n) = [r_1(\boldsymbol{X}_n), \ldots r_k(\boldsymbol{X}_n)]$, where k is the number of rules and element $r_j(\boldsymbol{X}_n) = 1 \Leftrightarrow r_j$ is satisfied by $\boldsymbol{X}_n$.

- **Prototype learning module:** The input encoded by rules and CNN embeddings are combined to form prototypes, $\boldsymbol{P} = \{\boldsymbol{p}_1, \ldots \boldsymbol{p}_k\}$, defining each rule in the high-dimensional space of CNN embeddings. Next, the prototypes are used to generate a

normalized similarity index for each epoch, $\boldsymbol{X}_n$, with each rule, r_j. These similarity indices are used to train an interpretable classifier such as decision trees or logistic regression.

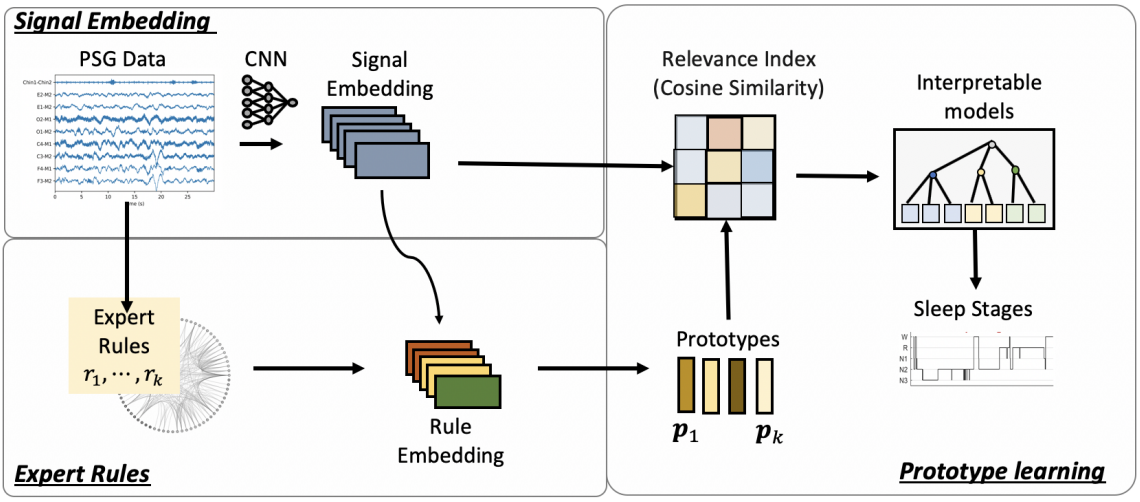


Figure 2.1: The SLEEPER framework.

2.2 Dataset

To evaluate the performance of SLEEPER, we conducted experiments using two datasets.

MGH This refers to a dataset containing PSG recordings of 2000 subjects from Massachusetts General Hospital. The MGH Institutional Review Board approved retrospective analysis of the clinically acquired data without requiring additional consent. The data were randomly selected from a mixture of diagnostic and split night recordings collected from patients whose ages range from 42 years old to 64 years old, with an average age of 53.

ISRUC This refers to the publicly available ISRUC data [22]. The ISRUC dataset contains PSG recordings of 100 subjects with evidence of having sleep disorders from the Sleep Medicine Centre of the Hospital of Coimbra University (CHUC). The data was collected from 55 male and 45 female subjects, whose ages range from 20 years old to 85 years old,

with an average age of 51.

Table 2.1: Dataset Summary

	MGH	**ISRUC** [22]
Number of PSGs	2,000	100
Number of Annotators	1	2
Channels Used	9	9

The EEG signals used for sleep staging are acquired from electrodes placed on the head as shown in Figure 2.2. Cz and Fz are the reference electrodes for the other channels. For contralateral referencing in our datasets, A1 and A2 are the reference electrodes, also called M1 and M2 respectively.

The recordings from both datasets were segmented into epochs of 30 seconds and visually scored by sleep technologists according to the guidelines of AASM [7]. The PSGs of both datasets include six EEG channels (F3, F4, C3, C4, O1, and O2), two Electrooculography (EOG) channels (E1 and E2) and a single Electromyography (EMG) channel, each referenced to the contralateral mastoid referred to as M1 and M2, or A1 and A2. Additionally, the ISRUC dataset includes scores by *two* sleep technologists. We can thus compare the agreement level between two experts and that between an expert and our algorithms.

2.3 Expert Rule Embedding

The majority of the rules in the guideline for sleep technicians [7] are vague. For example, LAMF, Low Amplitude Mixed Frequency is shown through multiple visual examples representing samples of the time domain signal but does not specify a threshold for low amplitude or the power distribution across different frequencies. As a result, it is not possible to computationally implement those rules with certainty.

Our approach alleviates the need for discrete boundaries by creating clusters based on

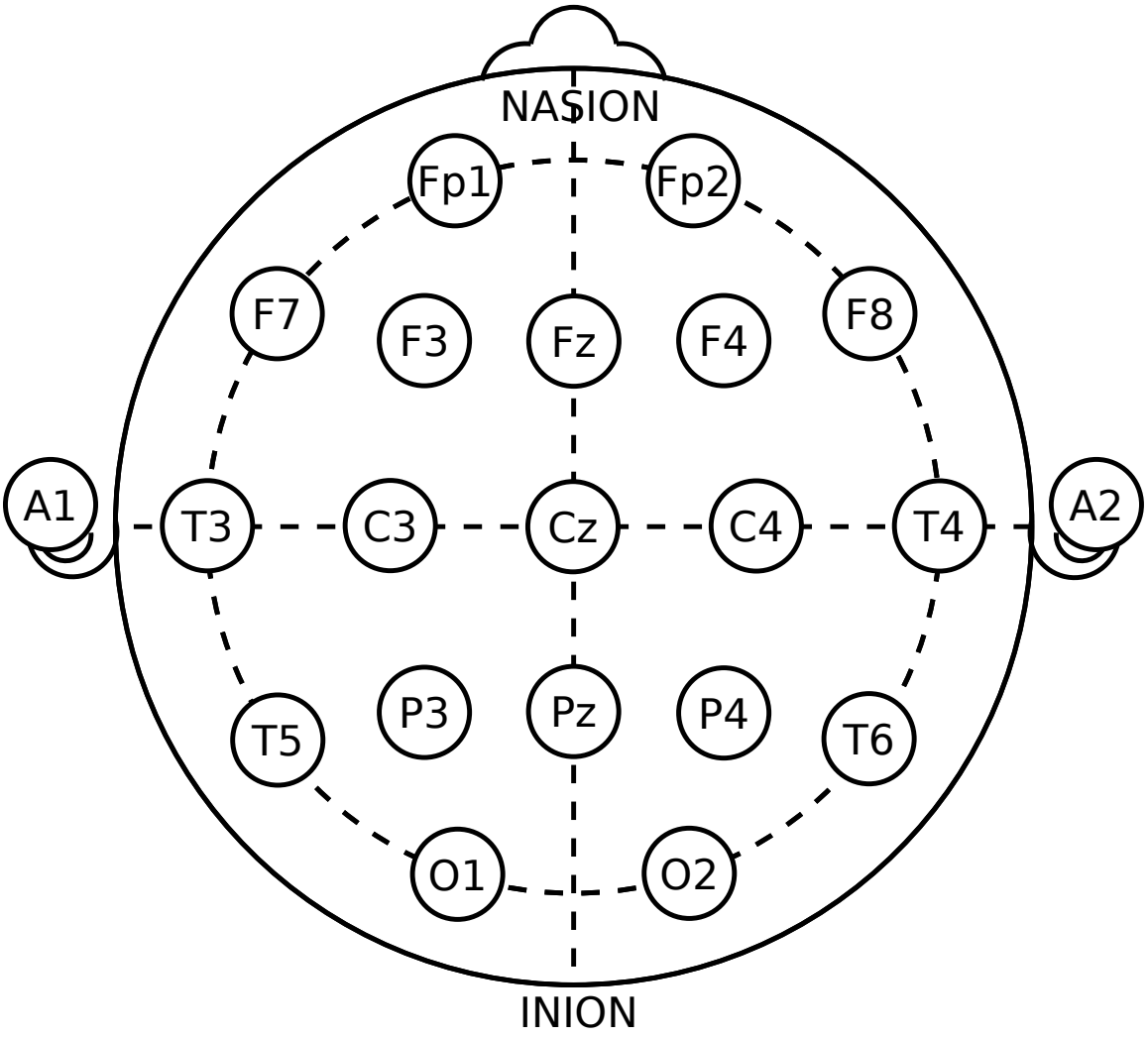


Figure 2.2: Location of electrodes for EEG in the 10–20 system.

the density of the features in each epoch. We incorporate expert suggestions to supplement the technical guidelines in the AASM manual [7]. Using this rule augmentation procedure, a set of 240 rules, $\boldsymbol{R}' = \{r'_1, \ \ldots \ r'_{240}\}$, are defined. Note that those rules are not directly associated with sleep stage labels like the AASM training manual. Instead, the rules define meaningful phenotypes, and the similarity with these phenotypes are used as input features to train sleep stage classifiers later, which lead to robust predictions.

The underlying features in those rules are described below, along with the channels utilized for each feature and the corresponding clustering scheme:

Sleep spindles are bursts of oscillatory signals originating from the thalamus and depicted in EEG [23]. It is a discriminatory feature of N2. We used the method proposed by [24] and [25] to extract spindles from contralateral signal pairs resulting in: (1) Number of sub-rules: 3 channel pairs with 4 groups for each channel; (2) Channel pairs: i. F3 & F4, ii. C3 & C4, iii. O1 & O2, and (3) Groups: i. $>$ 3s, ii. $>$ 6s, iii. $>$ 12s, iv. $>$ 18s in an epoch. In total, we have $3 \times 4 = 12$ binary features for spindles. For example, if both F3 and F4 channels exhibit greater than 12 seconds of spindles, the corresponding group will have a feature value of 1, and 0 otherwise.

Slow wave sleep (SWS) are distinguished by low-frequency and high-amplitude delta activity. Slow waves are the defining characteristics of N3. We utilize the method proposed by [26], [27], and [25] to extract SWS from contralateral signal pairs, including (1) Number of sub-rules: 3 channel pairs with 4 groups for each channel, (2) Channel pairs: i. F3 & F4, ii. C3 & C4, iii. O1 & O2, and (3) Groups: i. $>$ 3s, ii. $>$ 6s, iii. $>$ 12s, iv. $>$ 18s in an epoch.

Delta, Theta, Alpha, and Beta are the frequency bands that play differing roles in sleep staging. Delta (0.5-4Hz) waves delineate N3, Theta (4-8Hz) features in N1, Alpha (8-

12Hz) and Beta (>12Hz) discriminates between Wake and N1. The four bands in EMG determine the muscle tone used to distinguish between REM and Wake. We find the Power Spectral Density (PSD) using a multitaper spectrogram [28, 29] in each frequency band and make groups based on the percentile of PSD in the training dataset. (1) Number of sub-rules in each band: 9 channels with 4 groups for each channel, (2) Channels: i. F3, ii. F4, iii. C3, iv. C4, v. O1, vi. O2, vii. E1, viii. E2, ix. Chin EMG, and (3) Groups: i. $<$ 20th percentile, ii. $<$ 40th percentile, iii. $<$ 60th percentile, and iv. $<$ 80th percentile. In total, we have 6×4 binary features for each frequency band. For example, if the PSD of F3 across the Alpha band of an epoch is $<$ 20 th percentile the corresponding group will have feature value 1, otherwise 0.

Amplitude is important in discriminating Wake, REM, N1, and N2. Features used in sleep staging that are marked by distinctive amplitude include K Complexes, Chin EMG amplitude, Low Amplitude Mixed Frequency (LAMF). Since the AASM manual [7] does not declare concrete thresholds, we make groups for each and allow our decision tree to connote significance: (1) Number of sub-rules: 9 channels with 4 groups for each channel, (2) Channels: i. F3, ii. F4, iii. C3, iv. C4, v. O1, vi. O2, vii. E1, viii. E2, ix. Chin EMG, (3) Groups: i. $<$ 20th percentile, ii. $<$ 40th percentile, iii. $<$ 60th percentile, and iv. $<$ 80th percentile.

Kurtosis denotes the distribution of epochs. Although it is not directly related to any feature used by sleep experts, it helps detect outliers in data such as K Complexes which are rare events marked by a distinctive peak and trough. (1) Number of sub-rules: 9 channels with 4 groups for each channel, (2) Channels: i. F3, ii. F4, iii. C3, iv. C4, v. O1, vi. O2, vii. E1, viii. E2, ix. Chin EMG, (3) Groups: i. $<$ 20th percentile, ii. $<$ 40th percentile, iii. $<$ 60th percentile, and iv. $<$ 80th percentile.

Phenotype selection We analyze the efficacy of expert defined rules using ANOVA test and select the most discriminative rules. This reduces the number of expert rules from 240 to 96, where $\boldsymbol{R} = \{r_1, \dots r_{96}\}$ and $\boldsymbol{R} \subset \boldsymbol{R}'$. The resulting channels, underlying feature and the number of groups in each feature-channel pair are shown in Table 3.3. The results from applying all 96 rules on N epochs lead to a binary **rule assignment matrix** $\boldsymbol{R}(\mathcal{X}) \in \mathbb{R}^{N \times 96}$, which forms the basis of the interpretation module of `SLEEPER` framework.

$$\boldsymbol{R}(\mathcal{X}) = \begin{pmatrix} r_1(\boldsymbol{X}_1) & r_2(\boldsymbol{X}_1) & \dots \\ r_1(\boldsymbol{X}_2) & \ddots & \\ \vdots & & r_{96}(\boldsymbol{X}_N) \end{pmatrix} \tag{2.1}$$

where element $r_j(\boldsymbol{X}_i) = 1 \Leftrightarrow$ epoch $\boldsymbol{X}_i$ satisfies rule r_j, $\boldsymbol{X}_i \in \mathcal{X}$, and $\boldsymbol{X}_i \in \mathbb{R}^{9 \times 6{,}000}$. These resulting features are further discussed in Section 3.3.

2.4 Signal Embedding Generation

The multivariate time series PSG signals were embedded using CNN for capturing translation invariant and complex patterns. The network is composed of 3 convolutional layers as shown in Figure 2.3. Each convolutional layer is followed by ReLU activation and max pooling. By using a kernel size of 201, the convolutions in the first layer extract features based on 1 second segments of the multivariate time series data.

The output of the final convolutional layer, once flattened, is a vector $\boldsymbol{h}(\boldsymbol{X}_i) \in \mathbb{R}^{2{,}496}$. This is followed by a single fully connected layer with softmax activation to predict five different sleep stages:

$$\begin{aligned} \boldsymbol{z}_i &= \boldsymbol{W}^\top \boldsymbol{h}(\boldsymbol{X}_i) + \boldsymbol{b} \\ \boldsymbol{s}_i &= \text{softmax}(\boldsymbol{z}_i) \end{aligned}$$

where $\boldsymbol{W} \in \mathbb{R}^{2{,}496 \times 5}$ is the weight matrix, $\boldsymbol{b} \in \mathbb{R}^5$ is the bias vector, and $\boldsymbol{s}_i$ is the estimated

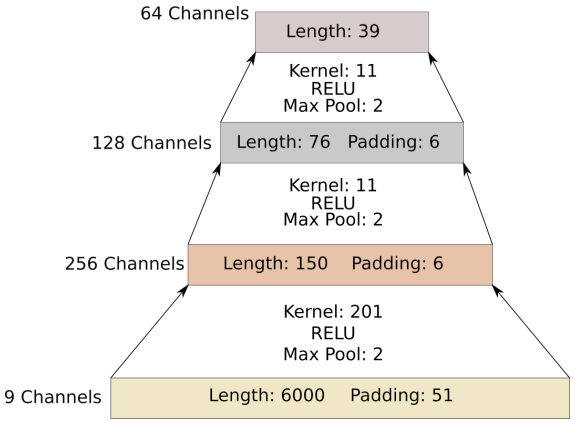


Figure 2.3: Convolution Layers of CNN

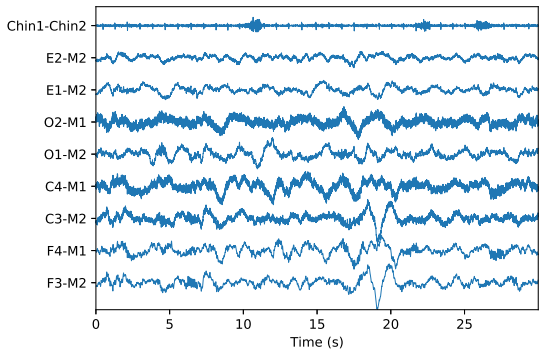


Figure 2.4: A 30 second epoch having 9 channels

probabilities of all 5 sleep stages at epoch i. To train the model, we used cross entropy loss in Eq. 2.2:

$$L(\boldsymbol{y}_i,\ \boldsymbol{s}_i) = -\sum_{j}^{5} \boldsymbol{y}_i[j]\ log(\boldsymbol{s}_i[j]) \tag{2.2}$$

where $L(\boldsymbol{y}_i,\ \boldsymbol{s}_i)$ is the estimated cross entropy loss for epoch i between human labels $\boldsymbol{y} \in \mathbb{R}^5$ and the predicted probabilities $\boldsymbol{s} \in \mathbb{R}^5$. After training on sleep stage prediction, we take the latent representation $\boldsymbol{h}(\boldsymbol{X})$ of 2,496 dimensions as the PSG signal embedding.

2.5 Prototype Learning and Relevance Matching

Each of the 96 rules leads to an embedding representation $\boldsymbol{p}_j \in \mathbb{R}^{2,496} | j = 1, 2, \ldots, 96$. Next we describe how to construct the embedding of prototypes using the latent representation of all epochs $\boldsymbol{h}(\mathcal{X})$. Once we have the rule assignment matrix $\boldsymbol{R}(\mathcal{X})$, each prototype representation $\boldsymbol{p}_j$ corresponds to the sum of latent embeddings of all the epochs that satisfy the rule j. Mathematically, all the prototypes can be computed as

$$\boldsymbol{P} = \boldsymbol{h'}(\mathcal{X})^T\ \boldsymbol{R}(\mathcal{X}) \tag{2.3}$$

where $\boldsymbol{P} \in \mathbb{R}^{2,496\times 96}$ is all the prototype embeddings, $\boldsymbol{h'}(\mathcal{X}) \in \mathbb{R}^{N\times 2,496}$ the column normalized representation of embedded input $\boldsymbol{h}(\mathcal{X})$[1].Next, we use cosine similarity in the embedding space to rank the similarity of any epoch with rules.

$$c_{i,j} = \frac{\boldsymbol{h}(\boldsymbol{X}_i)^\top \boldsymbol{p}_j}{||\boldsymbol{h}(\boldsymbol{X}_i)||_2 \; ||\boldsymbol{p}_j||_2} \tag{2.4}$$

where $c_{i,j} \in [0, 1]$ is the cosine similarity between the ith epoch and jth phenotype and $\boldsymbol{C}(\boldsymbol{h}(\mathcal{X})|\boldsymbol{P}) \in \mathbb{R}^{N\times 96}$.

We use these cosine similarity scores to all prototype embeddings as the input features to simple classifiers. We then train simple and interpretable classifiers such as shallow decision tree and logistic regression to provide the final classifications. When a new PSG, $\boldsymbol{X}_{test}$ is given, we find its latent representation using the trained CNN model $\boldsymbol{h}(\boldsymbol{X}_{test})$, followed by its cosine similarity to existing rule prototypes $\boldsymbol{C}(\boldsymbol{h}(\boldsymbol{X}_{test})|\boldsymbol{P})$. Using simple classifiers such as the decision tree, we obtain the predicted sleep stages.

[1]We empirically compared different normalization schemes and this column normalization led to the best performance in our tasks.

CHAPTER 3

EXPERIMENTS

3.1 Experiment Setting

Implementation Details. We implemented SLEEPER in PyTorch 1.0 [30] and scikit-learn [31]. We train the model using a machine equipped with Intel Xeon e5-2640, 256GB RAM, eight Nvidia Titan-X GPU, and CUDA 10.0. While training the CNN, we use a batch size of 1 PSG and ADAM as the optimization method. We train the CNN for 40 epochs. We set the learning rate at 10^{-4} and divide the learning rate by 10 once after 10 epochs.

To train the model, we randomly split the data by subjects into training and testing in a 9:1 ratio. For each dataset, we train using the training set to fix model parameters and test on the testing set for performance comparison. To ensure consistent performance across different datasets, we use the same model hyperparameters and underlying feature extraction schema to test both datasets. To evaluate SLEEPER, we consider the following baselines and evaluation metrics.

Baselines. We compare SLEEPER with the following baseline models on both datasets:

- **Convolutional Neural Network (CNN)** is the blackbox model used in obtaining the signal embeddings. It serves as the upper bound for performance.

- **Rules with Interpretable Classifier** where each epoch is represented by a multi-hot encoded binary rule assignment vector $\boldsymbol{R}(\boldsymbol{X}_i)$ from eqn. 2.1 and classification using gradient boosting (GB), decision tree (DT), and logistic regression (LR), respectively. For the choice of interpretable model in SLEEPER, we also consider DT, LR, and GB.

- **Mimic learning** [32] where the soft labels from a Recurrent Convolutional Neural

Network (RCNN) are used instead of the original hard labels and a gradient boosting regressor is then trained with those soft labels.

Additionally, on ISRUC dataset we also compare with the following baselines across 5 different sleep stages: (1) Agreement between two sleep experts on the same PSG recordings; (2) Maximum Overlap Discrete Wavelet Transform (MODWT) [22, 33]; (3) Logistic Smooth Transition Autoregressive (LSTAR) [34] (4) Convolutional Neural Network (CNN) [35] (5) RCNN on Spectrogram [10]. Note that (1) and (2) are conducted on the same ISRUC dataset, while (3-5) are on different datasets, which are only for a rough comparison.

Metrics. We compared testing performance using the following metrics, including accuracy (Acc), area under the receiver operator characteristics curve (ROC-AUC), and Cohen's κ. Here, Cohen's κ considers the possibility of assigning the correct sleep stage through random guesses. According to [36], $\kappa > 0.81,\ 0.8 > \kappa > 0.61,\ 0.6 > \kappa > 0.41,\ 0.4 > \kappa > 0.21,\ 0.2 > \kappa > 0.01,\ \kappa < 0.01$, means almost perfect, substantial, moderate, fair, slight, less than chance agreement respectively. We compare the performance across the 5 sleep stages using confusion matrices and class-wise sensitivity ($Sens^{(k)}$), also known as recall. Given expert annotations, $\mathcal{Y}'$ and predicted stages, $\mathcal{Y}$ of size N, $k = \{1, 2, 3, 4, 5\}$ indicating the sleep stage,

$$Acc = \frac{|\mathcal{Y} \cap \mathcal{Y}'|}{N}, \quad Sens^{(k)} = \frac{\left|\mathcal{Y}^{(k)} \cap \mathcal{Y}'^{(k)}\right|}{|\mathcal{Y}'^{(k)}|}$$

$$\kappa = \frac{Acc - p_e}{1 - p_e}, \text{ where } p_e = \frac{1}{N^2} \sum_{k}^{5} \left|\mathcal{Y}^{(k)}\right| \left|\mathcal{Y}'^{(k)}\right|$$

and $|\mathcal{Y}'^{(k)}|$ ($|\mathcal{Y}^{(k)}|$)is the number of human (algorithm) labels from sleep stage k.

3.2 Results: Staging Performance

The experimental results are compared in Table 3.1. On both datasets, SLEEPER performs almost as accurately as the black-box neural network models. Although SLEEPER achieved a significant reduction in dimensionality, from $\mathbb{R}^{2,496}$ to $\mathbb{R}^{96}$, the difference in AUC-ROC, accuracy, and Cohen's κ to the black-box CNN is relatively small. Moreover, each of those 96 dimensions is interpretable. SLEEPER-Decision Tree provides a list of normalized indices of length equal to the depth of the tree to indicate similarity with meaningful rules.

Table 3.1: Model Evaluation.[a] DT: Decision Tree, LR: Logistic Regression, GBT: Gradient Boosting Trees, Rule: Binary Features from Rules, CNN: Convolutional Neural Network

Model	Accuracy (%)		ROC-AUC (%)		Cohen's κ	
	MGH	ISRUC	MGH	ISRUC	MGH	ISRUC
SLEEPER-DT	78.3	78.5	85.0	84.7	0.694	0.720
SLEEPER-LR	79.8	77.0	86.1	84.9	0.714	0.699
SLEEPER-GBT	78.8	80.1	85.4	86.0	0.700	0.741
Rule & DT	66.1	67.1	75.7	78.2	0.510	0.564
Rule & LR	65.3	69.1	75.0	79.0	0.498	0.593
Rule & GBT	65.8	69.3	75.3	78.8	0.508	0.594
Mimic learning - GBT	67.5	62.1	78.6	76.4	0.540	0.514
CNN	81.6	82.4	87.4	87.8	0.742	0.772

[a] 96 rules and the corresponding prototypes are used in Rule and SLEEPER respectively

The sensitivity in classifying each sleep stage is compared with baselines in Table 3.2. The confusion matrices of our results using ISRUC dataset are shown in Figure 3.1. Agreement between experts are shown in Figure 3.1a and SLEEPER using a decision tree in Figure 3.1b. N1 classification is particularly problematic even for human sleep experts. This is due to the significant overlap in underlying criteria with N2. Beyond N2, SLEEPER with Decision Tree surpasses the performance of the baseline automatic sleep staging algorithm [33] while also providing interpretation. It exceeds expert agreement by a significant margin for N3. Reasons for this are further discussed in Section 3.4.

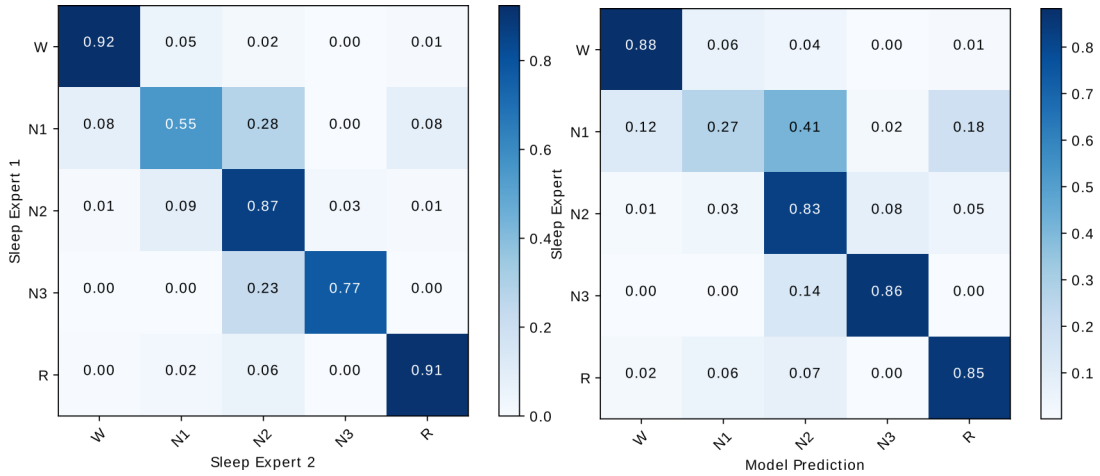
Table 3.2: Sensitivity across Sleep Stages [a]. DT: Decision Tree, GBT: Gradient Boosting Trees, LR: Logistic Regression

Model	Sensitivity (%)				
	Wake	REM	N1	N2	N3
SLEEPER-DT[b]	88.3	85.3	26.9	82.59	85.6
SLEEPER-LR	87.9	80.6	27.3	84.4	79.1
SLEEPER-GBT	88.1	86.1	34.5	83.2	87.9
Human Expert Agreement	92.4	91.2	55.4	86.6	77.4
MODWT [22]	88.3	81.8	39.3	80.2	83.5
CNN [35]	85	83	52	77	91
LSTAR [34]	88.7	88.4	50.3	85.0	87.4
RCNN on Spectrogram [10]	85	92	58	89	86

[a] The top 5 rows are results from the same cohort in ISRUC dataset [22]
[b] Depth, $D = 9$

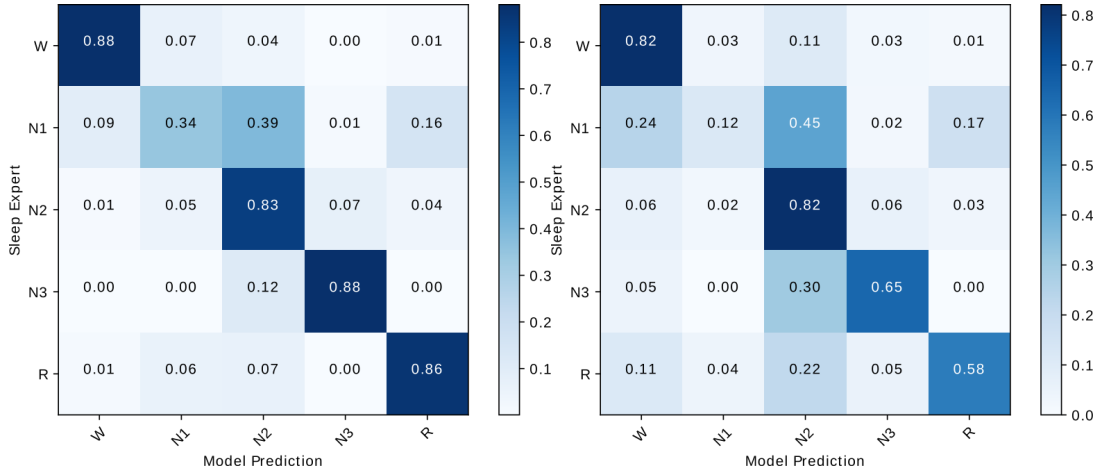
The agreement between two human experts in assigning sleep stages to our test PSGs in the ISRUC dataset is 83.0%, with a Cohen κ of 0.78. SLEEPER using a decision tree obtains an accuracy of 78.5%, with a Cohen's κ of 0.72 indicating substantial agreement according to the guidelines from [36].

Figure 3.2 shows the change in ROC-AUC of SLEEPER and rule-based method with the depth of the tree. It shows for trees of depth 9 we obtain ROC-AUC greater than 84% in both ISRUC and MGH Datasets. This indicates a group of 9 meaningful prototypes can classify the sleep stage in a sample well. Note, the larger size of MGH Dataset results in a much smoother distribution but the overall performance remains similar. This shows robustness across different datasets and the significant performance improvement from rules using SLEEPER.



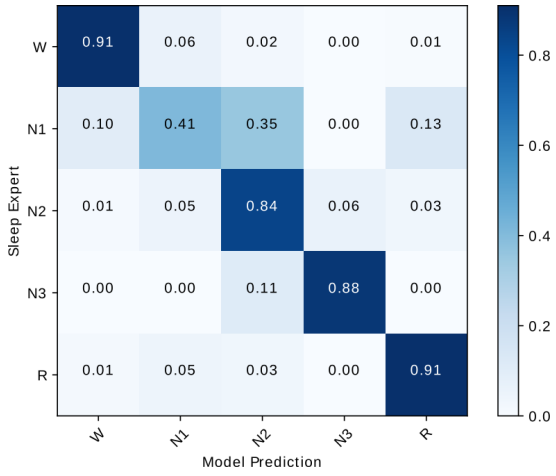
(a) Inter-Rater Agreement

(b) SLEEPER - Decision Tree



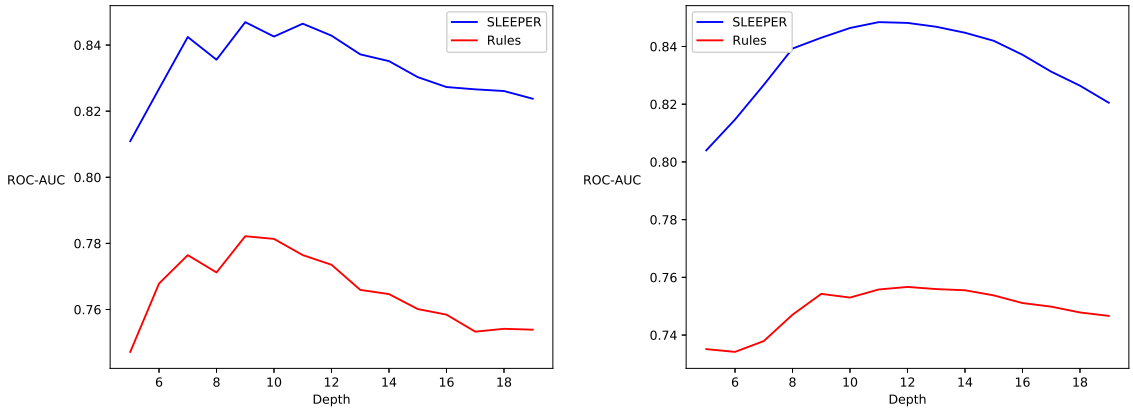
(c) SLEEPER - Gradient Boosting

(d) Rules and Decision Tree



(e) CNN

Figure 3.1: Confusion Matrices on ISRUC dataset



(a) ISRUC Dataset (b) MGH Dataset

Figure 3.2: SLEEPER ROC-AUC vs Tree Depth

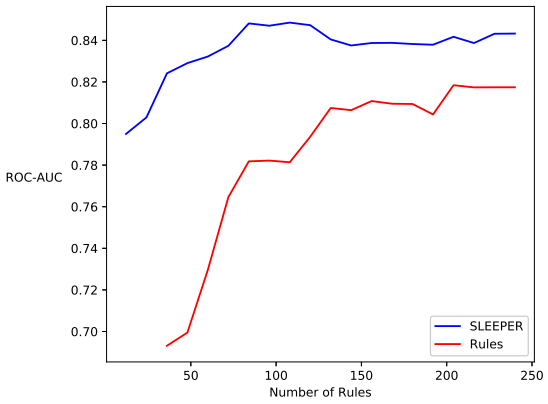


Figure 3.3: SLEEPER ROC-AUC vs Number of Rules

3.3 Selection of Expert Rules

Instead of handpicking due to ambiguity in their significance, we use analysis of variance (ANOVA) models to rank features by significance and reduce the number of rules from 240 to 96. The change in ROC-AUC with number of rules is shown in Figure 3.3. It reveals the discrepancy in performance between SLEEPER and the rule based method. The selected number of expert rules from each channel-feature pair is shown in Table 3.3. Note that the selected features do not include any features from Spindles and Kurtosis. In particular, sleep spindles have frequency range of 12-14 Hz with a duration of 0.5-1.5 seconds but their use in detecting N2 is practically difficult. This could be due to hidden spindles in

Table 3.3: 96 selected rules out of 240 expert rules using ISRUC dataset

Channels	Features							
	Spindle	SWS	Delta	Theta	Alpha	Beta	Kurtosis	Amplitude
F3-A2		3	4		4	4		4
F4-A1		3	4	4	4			4
C3-A2		2	4	4		4		4
C4-A1		2	4					4
O1-A2			4	4				4
O2-A1			4	3	4			4
ROC-A2			4					4
LOC-A2								
Chin EMG								

Table 3.4: 96 selected rules out of 240 expert rules using MGH Dataset

Channels	Features							
	Spindle	SWS	Delta	Theta	Alpha	Beta	Kurtosis	Amplitude
F3-M2		3	4	4	4	4		4
F4-M1		3	4	4	4			4
C3-M2		3	4	4				4
C4-M1		3	4					4
O1-M2		2	4	4				4
O2-M1		2	4	4				4
E1-M2					4			4
E2-M2								
Chin1-Chin2								

other stages [24]. And Kurtosis is a common statistical measure but is not directly related to key features described in sleep scoring manual. We also observe the removal of rules using EMG and second EOG channels. EMG recordings are particularly noisy with low amplitude, as shown in the top channel of Figure 2.4.

K Complex, another underlying feature in N2 and REM detection, contains a distinctive rise and fall which is larger than the amplitude of regular signal oscillations. K Complexes, unfortunately, are not reliable enough for our use case with an inter-rater κ of .51 [37]. On the other hand, amplitude-based prototypes play a big role in `SLEEPER`. Low Amplitude

Mixed Frequency (LAMF) is a feature used for discriminating Wake, N1, N2, and REM. The channels used in detecting LAMF are not mentioned in the guidelines [7].

3.4 Interpretation

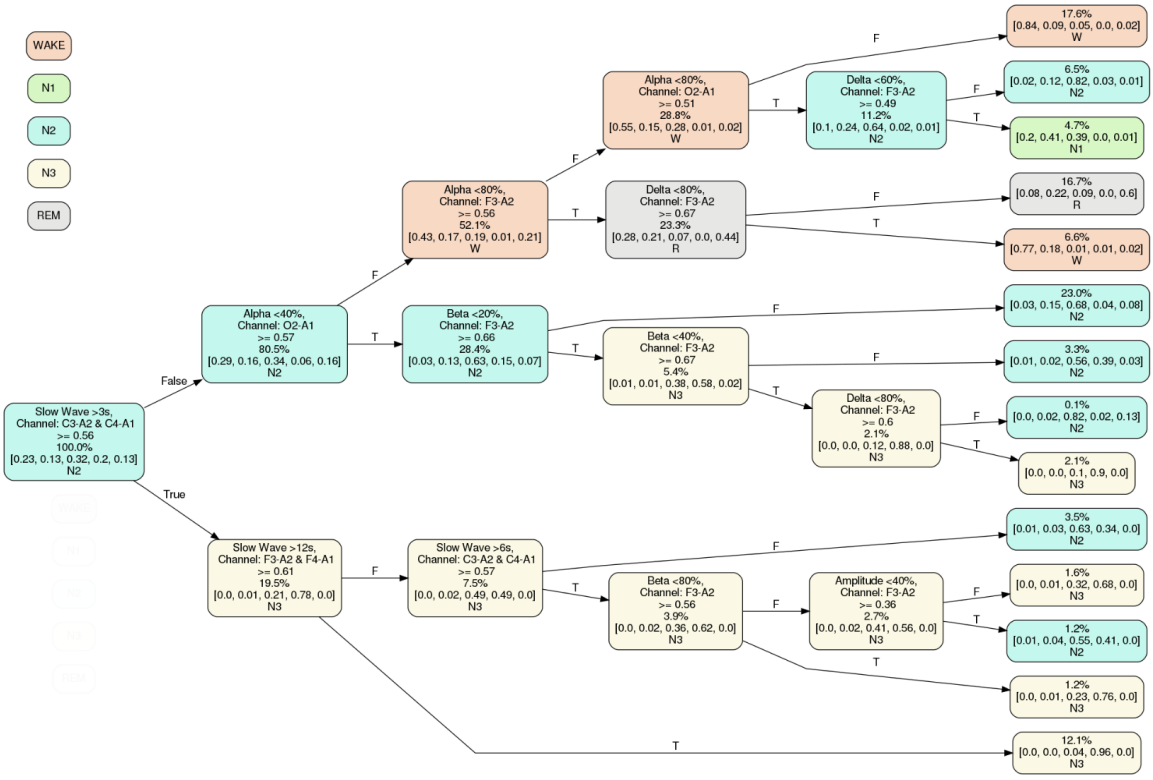


Figure 3.4: SLEEPER-Decision Tree trained from the ISRUC dataset

For example, the root node considers the prototype defining slow waves greater than 3 seconds (row 1) on both C3-A2 and C4-A1 channels (row 2). The cosine similarity threshold is 0.56 (row 3). 100% (row 4) of all training examples go through this node. [.23, .13, .32, .2, .13] (row 5) are the probabilities of the sleep stages in the training examples in the order of [Wake, N1, N2, N3, R] with N2 (row 6) as the majority.

Figure 3.4 shows a decision tree of depth, $D = 5$, based on SLEEPER. This obtains an AUC-ROC of 81%. The leftmost node denotes the root of the tree and the color indicates the most frequent sleep stage for training data passing through that node. The six rows of each node contain the following: (1) the underlying feature and grouping criteria, (2) the channels used to extract the feature, (3) the cosine similarity with the resulting rule, (4) the percentage of data passing through the node, (5) the ratio of each sleep stage in data

passing through the node in the following order: [Wake, N1, N2, N3, REM], (6) the most frequent sleep stage at the node, in other words, if classification is performed at that node we will assign this label. The leaves on the right contain the same contents as the lowest 3 rows at other nodes.

Analyzing the resulting decision tree reveals some promising aspects of SLEEPER. According to the sleep staging guidelines for human annotators [7], N3 is distinguished by the occurrence of slow waves. One of the underlying features of our rules is slow waves. We created 4 binary features based on the duration of slow waves in each 30s epoch, > 3s, > 6s, > 12s, and > 18s. The first node creates a split based on cosine similarity ≥ 0.56 with the prototype, Slow Waves of duration greater than 3s in the Central Channels. Since slow waves are predominant in N3, 78% of training data that satisfied the aforementioned criteria in the next node contains N3, while only 6% of the other child contains N3. The next node restricts the threshold to 12s in the Frontal region. 96% of the resulting leaf node classifying 12.1% of the training dataset was labeled, in agreement with SLEEPER, as N3 by experts.

Furthermore, analyzing the leaves, we observe stages with similar characteristics that occur in pairs, like REM and Wake, N3 and N2, N1, and N2. We notice that the top right leaf containing Wake is distinguished by Alpha activity in the Occipital Region. This criterion for detecting Wake is mentioned in the guidelines for human annotators [7].

So, we can see the inference logic in SLEEPER is meaningful and to a great extent corresponds to the rules stated in the AASM Manual [7].

CHAPTER 4
CONCLUSION

Interpretability and accuracy are often a trade-off to each other in machine learning modeling. Especially in the age of deep learning, many accurate models are black-box models that do not provide any insight into the reasoning behind predictions. On the other hand, simple models like decision trees often result in inaccurate predictors. In this thesis, we present a method that introduces a deep prototype learning method that provides accurate predictions as well as very simple and intuitive prediction models with a shallow decision tree. We develop and evaluate the methods in the context of sleep staging applications on PSG data from two sleep labs. The proposed method achieves performance metrics in sleep staging tasks comparable to the state of the art baselines based on deep learning. A qualitative case study illustrated a simple and intuitive decision tree that can perform accurate sleep staging classification while explaining interpretable rules. The interpretation shows similarity with the decision-making process used by human experts in the field and reveals the reasoning behind stages with low classification accuracy.

REFERENCES

[1] Z. C. Lipton, "The mythos of model interpretability," *Queue*, vol. 16, no. 3, 30:31–30:57, Jun. 2018.

[2] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[3] J. B. Stephansen, A. N. Olesen, M. Olsen, A. Ambati, E. B. Leary, H. E. Moore, O. Carrillo, L. Lin, F. Han, H. Yan, *et al.*, "Neural network analysis of sleep stages enables efficient diagnosis of narcolepsy," *Nature communications*, vol. 9, no. 1, p. 5229, 2018.

[4] A. C. Krieger, *Social and Economic Dimensions of Sleep Disorders, An Issue of Sleep Medicine Clinics, E-Book*, 1. Elsevier Health Sciences, 2017, vol. 12.

[5] O. K. Olsen, S. Pallesen, and E. Jarle, "The impact of partial sleep deprivation on moral reasoning in military officers," *Sleep*, vol. 33, no. 8, pp. 1086–1090, 2010.

[6] ASA, *Sleep statistics - research & treatments — american sleep assoc*, `https://www.sleepassociation.org/about-sleep/sleep-statistics/`, 2019.

[7] R. B. Berry, R. Budhiraja, D. J. Gottlieb, D. Gozal, C. Iber, V. K. Kapur, C. L. Marcus, R. Mehra, S. Parthasarathy, S. F. Quan, *et al.*, "Rules for scoring respiratory events in sleep: Update of the 2007 aasm manual for the scoring of sleep and associated events," *Journal of clinical sleep medicine*, vol. 8, no. 05, pp. 597–619, 2012.

[8] A. Sors, S. Bonnet, S. Mirek, L. Vercueil, and J.-F. Payen, "A convolutional neural network for sleep stage scoring from raw single-channel eeg," *Biomedical Signal Processing and Control*, vol. 42, pp. 107–114, 2018.

[9] H. Dong, A. Supratak, W. Pan, C. Wu, P. M. Matthews, and Y. Guo, "Mixed neural network approach for temporal sleep stage classification," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 2, pp. 324–333, 2018.

[10] S. Biswal, J. Sun, H. Sun, M. B. Westover, B. Goparaju, and M. T. Bianchi, "Expert-level sleep scoring with deep neural networks," *Journal of the American Medical Informatics Association*, vol. 25, no. 12, pp. 1643–1650, Nov. 2018. eprint: `http://`

oup.prod.sis.lan/jamia/article-pdf/25/12/1643/27090555/ocy131.pdf.

[11] M. Längkvist, L. Karlsson, and A. Loutfi, "Sleep stage classification using unsupervised feature learning," *Adv. Artif. Neu. Sys.*, vol. 2012, 5:5–5:5, Jan. 2012.

[12] Z. C. Lipton, "The mythos of model interpretability," *CoRR*, vol. abs/1606.03490, 2016. arXiv: 1606.03490.

[13] J. L. Kolodner, "An introduction to case-based reasoning," *Artificial intelligence review*, vol. 6, no. 1, pp. 3–34, 1992.

[14] C. E. Priebe, D. J. Marchette, J. G. DeVinney, and D. A. Socolinsky, "Classification using class cover catch digraphs," *Journal of classification*, vol. 20, no. 1, pp. 003–023, 2003.

[15] J. Bien and R. Tibshirani, "Prototype selection for interpretable classification," *The Annals of Applied Statistics*, pp. 2403–2424, 2011.

[16] B. Kim, C. Rudin, and J. A. Shah, "The bayesian case model: A generative approach for case-based reasoning and prototype classification," in *Advances in Neural Information Processing Systems*, 2014, pp. 1952–1960.

[17] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," *CoRR*, vol. abs/1703.05175, 2017. arXiv: 1703.05175.

[18] O. Li, H. Liu, C. Chen, and C. Rudin, "Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions," *CoRR*, vol. abs/1710.04806, 2017. arXiv: 1710.04806.

[19] I. Al-Hussaini, C. Xiao, M. B. Westover, and J. Sun, "Sleeper: Interpretable sleep staging via prototypes from expert rules," in *Proceedings of the 4th Machine Learning for Healthcare Conference*, F. Doshi-Velez, J. Fackler, K. Jung, D. Kale, R. Ranganath, B. Wallace, and J. Wiens, Eds., ser. Proceedings of Machine Learning Research, vol. 106, Ann Arbor, Michigan: PMLR, 2019, pp. 721–739.

[20] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[21] S. Schutte-Rodin, L. Broch, D. Buysse, C. Dorsey, and M. Sateia, "Clinical guideline for the evaluation and management of chronic insomnia in adults," *Journal of Clinical Sleep Medicine*, vol. 4, no. 05, pp. 487–504, 2008.

[22] S. Khalighi, T. Sousa, J. M. Santos, and U. Nunes, "Isruc-sleep: A comprehensive public dataset for sleep researchers," *Computer methods and programs in biomedicine*, vol. 124, pp. 180–192, 2016.

[23] L. De Gennaro and M. Ferrara, "Sleep spindles: An overview," *Sleep medicine reviews*, vol. 7, no. 5, pp. 423–440, 2003.

[24] K. Lacourse, J. Delfrate, J. Beaudry, P. Peppard, and S. C. Warby, "A sleep spindle detection algorithm that emulates human expert spindle scoring," *Journal of neuroscience methods*, vol. 316, pp. 3–11, 2019.

[25] R. Vallat, *Raphaelvallat/yasa: V0.1.3*, Mar. 2019.

[26] J. Carrier, I. Viens, G. Poirier, R. Robillard, M. Lafortune, G. Vandewalle, N. Martin, M. Barakat, J. Paquet, and D. Filipini, "Sleep slow wave changes during the middle years of life," *European Journal of Neuroscience*, vol. 33, no. 4, pp. 758–766, 2011.

[27] M. Massimini, R. Huber, F. Ferrarelli, S. Hill, and G. Tononi, "The sleep slow oscillation as a traveling wave," *Journal of Neuroscience*, vol. 24, no. 31, pp. 6862–6870, 2004.

[28] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, and M. S. Hämäläinen, "Mne software for processing meg and eeg data," *Neuroimage*, vol. 86, pp. 446–460, 2014.

[29] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, *et al.*, "Meg and eeg data analysis with mne-python," *Frontiers in neuroscience*, vol. 7, p. 267, 2013.

[30] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[32] Z. Che, S. Purushotham, R. Khemani, and Y. Liu, "Distilling knowledge from deep networks with applications to healthcare domain," *arXiv preprint arXiv:1512.03542*, 2015.

[33] S. Khalighi, T. Sousa, G. Pires, and U. Nunes, "Automatic sleep staging: A computer assisted approach for optimal combination of features and polysomnographic channels," *Expert Systems with Applications*, vol. 40, no. 17, pp. 7046–7059, 2013.

[34] P. Ghasemzadeh, H. Kalbkhani, S. Sartipi, and M. G. Shayesteh, "Classification of sleep stages based on lstar model," *Applied Soft Computing*, vol. 75, pp. 523–536, 2019.

[35] S. Chambon, M. N. Galtier, P. J. Arnal, G. Wainrib, and A. Gramfort, "A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 4, pp. 758–769, 2018.

[36] A. J. Viera, J. M. Garrett, *et al.*, "Understanding interobserver agreement: The kappa statistic," *Fam med*, vol. 37, no. 5, pp. 360–363, 2005.

[37] T. Lajnef, S. Chaibi, J.-B. Eichenlaub, P. M. Ruby, P.-E. Aguera, M. Samet, A. Kachouri, and K. Jerbi, "Sleep spindle and k-complex detection using tunable q-factor wavelet transform and morphological component analysis," *Frontiers in human neuroscience*, vol. 9, p. 414, 2015.